Rationalization of Goal Models in GRL using Formal Argumentation

Marc van Zee,* Floris Bex,† Sepideh Ghanavati‡,§
* Department of Individual and Collective Reasoning, University of Luxembourg, Luxembourg
† Department of Information and Computing Sciences, Utrecht University, The Netherlands
‡ Luxembourg Institute of Science and Technology (LIST), Luxembourg
§ School of Computer Science, Carnegie Mellon University
marc.vanzee@uni.lu, sepideh.ghanavati@list.lu, floris.bex@uu.nl

***Abstract*—We apply an existing formal framework for practical reasoning with arguments and evidence to the Goal-oriented Requirements Language (GRL), which is part of the User Requirements Notation (URN). This formal framework serves as a rationalization for elements in a GRL model: using attack relations between arguments we can automatically compute the acceptability status of elements in a GRL model, based on the acceptability status of their underlying arguments and the evidence. We integrate the formal framework into the GRL metamodel and we set out a research to further develop this framework.**

***Index Terms*—User Requirements Notation, Goal-oriented Requirements Language, goal modeling, formal argumentation**

I. Introduction

Goal modeling is an essential part of all development processes, especially in the area of complex reactive and distributed systems [16]. The Goal-oriented Requirements Language (GRL) [2] is part of the User Requirements Notation (URN) standard [13]. GRL aims to capture business or system goals, (sub)goals and tasks that help achieve high-level goals [1]. GRL allows for several types of analysis and evaluation techniques to assess the satisfaction of goals and thus to decide on high-level alternatives.

Although it is currently possible to rationalize GRL models through so-called *belief* elements, these beliefs are single and static statements. Such static elements do not capture the dynamic goal modeling process, nor do they capture the discussion process between stakeholders in which high-level softgoals are translated into subgoals, which are in turn translated into tasks [16]. Hence, the GRL model with the belief elements is merely an end product of this process, which does not reflect how the model was created, i.e., what reasons were used to choose certain elements in the model and to reject the others and what evidence was given as the basis of this reasoning.

In this paper, we develop an initial framework for tracing elements of a GRL model to discussions between stakeholders, captured as arguments and counterarguments based on evidence. The framework is based on the formal argumentation framework $ASPIC^+$ for structured argumentation [15], extended with ideas on practical argumentation [3] and argumentation based on evidence [5]. The formal semantics [7] of arguments and counterarguments underlying the argumentation framework allow us to determine whether the elements of a GRL model are acceptable given the potential contradictory evidence and stakeholders' opinions. Thus, we add a new formal evaluation technique for goal models that allows us to assess the *acceptability* of elements of a goal model (as opposed to the *satisfiability* [1]).

Other authors, most notably Jureta et al. [14], have applied formal argumentation methods to goal rationalization. Our framework shares many of the basic ideas of this other work. However, in section VI we motivate that in our research agenda we significantly extend this earlier work, offering a richer, more standardized and fully implemented framework.

The structure of this paper is as follows: Section II provides a simple fictitious running example that illustrates the key elements of our framework. In Section III we present our main technical contribution: the formal framework for practical reasoning based on arguments, and in Section IV we show the starting point of the implementation of this framework into the GRL metamodel. We set out our research agenda for future work in Section V, and in Section VI we discuss related work.

II. Running Example: Best Furniture Inc.

Best Furniture Inc. is a fictitious company that builds and sells furniture. They recently noticed a decrease of income from store purchases. To this end, the stakeholders are discussing how to adapt their business processes in order to make shopping at the store more attractive, and in this way increase the revenue.

A CRM expert argues that improving customer support for store purchases will most likely lead to an increase of profit, simply because customers will be more satisfied. While thinking of ways in which customer support may be improved, one of the stakeholders mentions that they have a very weak policy for returning products: It is currently not possible to return products after buying them. A sales clerk confirms that there have been regular complaints about this policy. Another stakeholder disagrees and points out that Best Furniture Inc. should be careful with such a return policy, because if too many customers return their products it may in fact cost the company a lot of money. However, the stakeholder in favor of the return policy is able to produce data from other furniture companies showing that customers tend to keep the furniture that they have, and therefore there will not be many returns.

After this discussion, the stakeholders decide to implement the possibility for customers to return their products. The stakeholders agree that if a customer would like to return a product, then the following three conditions should be satisfied: First, The product is bought from company "Best Furniture Inc.". Secondly, the customer has a receipt for the product, and thirdly, the product is undamaged.

We assume familiarity with the basics of goal modeling and GRL, and we refer to [1] for a more detailed overview. A GRL model of our example is depicted in Figure 1. The goal ( ) Implement Return Policy contributes positively to the softgoal ( ) Improve Customer Support, which in turn contributes positively to the softgoal Increase Profit. The goal Implement Return Policy is decomposed into three tasks ( ) Check for receipt, Check if item is bought, and Check for damage. The rationalization of the choice to implement a return policy is provided by attaching a belief element ( ) to the corresponding GRL element.

Comparing the GRL model to the description of the discussion between the stakeholders shows that although the GRL model is able to model the goals, subgoals and tasks, it does not capture the stakeholders' arguments. For example, there was an extensive discussion about the goal Implement Return Policy: one stakeholder argued that it was too expensive, but another stakeholder countered this by saying that not many people return furniture.

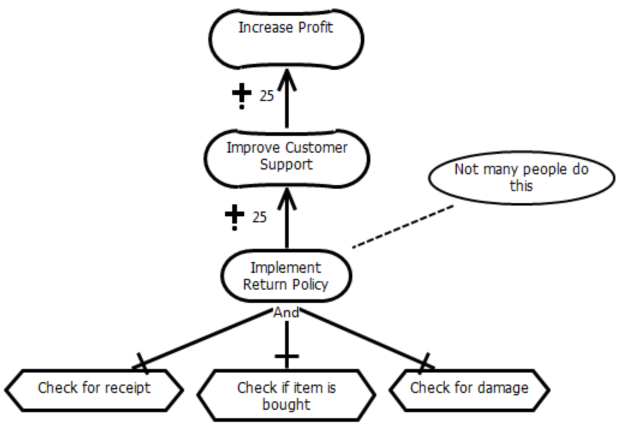

Fig. 1: GRL Model of the running example

A further shortcoming of the GRL model is that it does not capture the evidence given for various elements of the GRL model. For example, the CRM expert argued that increasing customer support leads to more profit. This claim is represented by the positive link between Improve Customer Support and Increase Profit, but the way the evidence contributes to, for example, the strength of this link is not captured in the model.

III. A Framework for Goal-Based Argumentation with Evidence

Reasoning about which goals to pursue and actions to take is often referred to as *practical reasoning*. Practical reasoning has been studied extensively in formal argumentation, most notably by Atkinson *et al.* [3], who define the following basic argument structure for reasoning from goals to actions.

$$\begin{aligned} &\text{I have goal } G \\ &\text{Doing actions } A \text{ will realize goal } G \\ &\textit{Therefore} \text{ I should do actions } A. \end{aligned} \tag{1}$$

This basic argument can be further extended to capture subgoals (*i.e.*, realizing goals $G_1, \ldots, G_n$ will allow me to realize goal G_i).

Practical reasoning is defeasible, in that conclusions which are at one point acceptable can later be rejected because of new information or claims. For example, there might be unwanted side effects to performing action A, or there may be alternative actions that also lead to the realization of one's goals. Atkinson *et al.* [3] define a formal set of critical questions that point to typical ways in which a practical argument can be criticized. Our assumption is that the dynamic discussions about goals and actions in GRL can be captured using practical argumentation.

Arguing about goals and actions also contains an element of *epistemic reasoning*, as people might disagree on whether action A really realizes goal G, or on whether goal G is actually a valid goal, and so on. The earlier example of the CRM expert, for instance, is about the (causal) link between increasing customer support and the increase of profits. In GRL, these links are captured by the contribution and correlation links and, whilst it is possible to indicate a quantitative value, it is impossible to argue about them, or support them with evidence. In a practical argument, such links are captured by one of the premises - i.e. 'doing action A will realize goal G' or 'realizing goals $G_1, \ldots, G_n$ will allow me to realize goal G_i' - and we can use further arguments to reason about these premises, supporting or countering them with evidence (cf. Bex et al. [5]). For example, we can use the CRM expert's statements as support for the link between customer support and profits, or counter the expert's statements with an argument based on data that shows no increase of profits for increased customer support.

In order to rationalize and formally evaluate our goals and beliefs using argumentation, we choose the *ASPIC*$^+$ framework for structured argumentation [15]. This framework allows us to define our own inference rules for practical and epistemic reasoning inspired by [3] and [5] and, since *ASPIC*$^+$ is an instantiation of Dung's [7] abstract framework, we can use the standard argumentation calculus to compute whether we are justified in accepting certain goals in our GRL diagram given the criticisms and evidence that have been brought forward.

A. *The* ASPIC$^+$ *Framework*

In recent decades, the computational study of argumentation has received increasing attention, especially following the influential paper of Dung [7] on argumentation semantics. The basics of computational argumentation stem from classical logic: given a set of premises (expressed in a logical language) and a set of inference rules, we can infer a conclusion. The *ASPIC*$^+$ framework captures these elements in the definition of an *argumentation theory*, which consists of a *logical language*

L, a *knowledge base* K and a set R of *inference rules*. We discuss each of these three elements in turn.

In our case, the *logical language* L is propositional logic with modalities G for *goals*[1], B for beliefs, A for actions, and E for evidence, which allows us to express the GRL elements. The language is further extended with a connective *contributes_to* for expressing the contribution and decomposition links of GRL. For instance, the goal from our running example to improve customer support is expressed by G(`improve_cs`), while B(`few_product_returns`) expresses the belief that few people will return products, A(`check_receipt`) expresses the action of checking the receipt, and E(`CRM_expert`) expresses evidence provided by a CRM expert. Two examples of *contributes_to* connectives from our running example are: "`return_product` *contributes_to* `improve_cs`" and " `receipt` ∧ `bought` ∧ `damaged` *contributes_to* `return_product`". The first expression states that returning a product for money contributes to improving customer support. The second expression shows how decomposition links can be captured using conjunctions: if the customer has a receipt, the item was bought and not damaged, then the product can be returned.

The *knowledge base* K specifies the body of information from which the premises of an argument can be taken. A distinction is made between ordinary premises (K_p), which are uncertain assumptions and can be attacked by other arguments, and axioms (K_a), i.e. certain premises, that cannot be attacked. Part of the knowledge base K of our running example can be formalized as follows:

K_a: {E(`CRM_expert`), E(`sales_clerk`)}
K_p: {G(`increase_profit`),
c_1 : `improve_cs` *contributes_to* `increase_profit`,
c_2 : `return_product` *contributes_to* `improve_cs`}

Thus, the evidence `CRM_expert` and `sales_clerk` are the axioms of the knowledge base, meaning that they cannot be attacked. The *contributes_to* formulas have been named c_1 and c_2 for convenience. Note that c_1 and c_2 are both directly taken from the contribution and decomposition links in GRL from Figure 1.

The set R of inference rules consists of two different types of inference rules: deductive or *strict* inference rules (R_s) are those of propositional logic and permit deductive inferences from premises to conclusions; and *defeasible* inference rules (R_d) represent uncertain inferences that can be attacked. Defeasible inference steps are of the form $A \mid\!\sim b$ (from a set of premises A we can defeasibly infer b).

We introduce two new defeasible inferences rules for practical arguments.

PA   Gq, a *contributes_to* $q \mid\!\sim$ Aa

Here, a is an action and q corresponds to a goal. Thus, if q is a goal and doing action a contributes to q, then we should perform a.

[1]Note that in our framework we do not distinguish between softgoals and hard goals.

PG   Gq, p *contributes_to* $q \mid\!\sim$ Gp

This inference rule expresses that if q and p are goals, and realizing p contributes to q, then we should make p a goal. In addition to these two rules for practical reasoning, we can also formalize inferences based on evidence through specific defeasible rules. For example, the statement given by the CRM expert (section II) is a reason to believe that the returning a product improves customer support is formalized as follows:

EV$_1$ E(`CRM_expert`) $\mid\!\sim$ `improve_cs` *contributes_to* `increase_profit`

Similarly, the statement by the company's sales clerk – that implementing an option to allow people to return products will improve customer service – can be formalized as follows:

EV$_2$ E(`sales_clerk`) $\mid\!\sim$ `return_product` *contributes_to* `improve_cs`

B. Building Practical Arguments

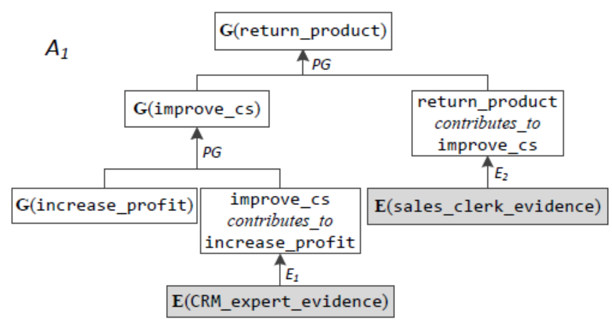

Fig. 2: An argument based on the GRL in figure 1

In our framework, an argument is a tree. The leaves of this tree (i.e. the argument's premises) are elements of the knowledge base K and all other nodes are inferred using an inference rule in R. The root of the tree is the argument's conclusion, and subtrees are called subarguments. As an example, assume that we have strict inference rules from propositional logic and defeasible inference rules PA, PG, EV$_1$ and EV$_2$ discussed above.

Given these inference rules and the set K, we can construct the argument in Figure 2. In this figure, the arrows stand for inferences, where the inference rule is indicated next to the arrow. Note that argument A_1 can be extended into arguments for G(`bought`) or G(¬`damaged`) using the *contributes_to* connective "`return_product` *contributes_to* `improve_cs`".

Since the conclusion of argument A_1 is G(`return_product`), using the PA inference rule we can then further discuss which actions we have to take to realize these goals. There is, then, an exact correspondence between the GRL model from Figure 1 and these arguments: the goals, actions and links in GRL are explicitly represented as elements of the argument.

C. Attacks between Arguments

An important feature of argumentation is that arguments can be attacked by counterarguments. A common way to attack an argument is to contradict one of its (sub)conclusions. For example, it was argued that allowing customers to return products would cost the company a lot of money. This argument can be formalized as A_2 in Figure 3[2]. The argument A_2 attacks the original argument A_1 from Figure 2 and vice versa (rendered as dashed lines with an open arrowhead): G(`return_product`) and ¬G(`return_product`) are obviously contradictory. In the discussion, however, argument A_2 was countered with evidence of statistics (`stats_evidence`) showing that few people return products (`few_product_returns`). This allows us to construct argument A_3, which attacks A_2, because we can say that the fact that there will be few returns contradicts the belief that allowing such returns will cost a lot of money.

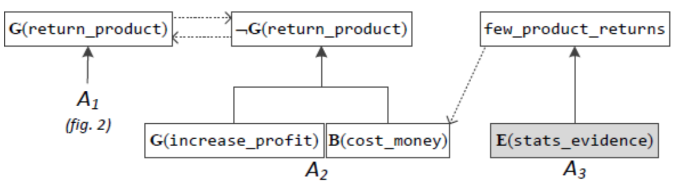

Fig. 3: Attacking arguments

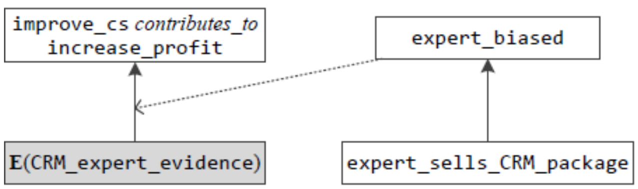

Fig. 4: Providing an exception to an inference

Another way to attack an argument is to deny an inference step by giving an exception to an inference rule. For example, whilst normally we would consider expert statements to be good reasons for believing something (cf. inference rule EV_1), an exception to such an "inference from expert opinion" occurs when we have an argument that the expert is in fact biased (for example, because he would like to sell a specific CRM package that facilitates product returns). Figure 4 shows this counterargument to the expert argument. Note that not the premises nor conclusion is attacked, but rather the inference itself.

D. The Acceptability of Arguments

Given a collection of arguments and their attack relations, we can use various argumentation semantics [7] to determine the *acceptability* of the arguments. Such semantics abstract away from the internal structure of the arguments and consider only arguments and their attacks. Informally, all arguments that are not attacked by any argument are *IN*. Arguments that are attacked by an argument that is *IN* are *OUT*, and arguments that are only attacked by arguments that are *OUT* are *IN* as well. Otherwise, an argument is *UNDECIDED*. We have visualized the arguments of our running example in Figure 5, where A_1 and A_3 are *IN*, and A_2 is *OUT*. Note that if argument A_3 would not exist, both A_1 and A_2 would be UNDECIDED.

Fig. 5: The acceptability of arguments

Since the elements of the arguments correspond to the elements of the GRL model, we can determine the status of these GRL elements based on the underlying rationalizations, which are made explicit in the argumentation theory. In this way, the argumentation theory provides a formal grounding for GRL, since we can apply Dung's [7] calculus for determining the status of arguments to elements of GRL. Thus, in our example, the goal G(`return_product`) is *IN*, while the goal ¬G(`return_product`) is *OUT* (cf. the the GRL model in Figure 1).

IV. The Metamodel

Figure 6 depicts the metamodel linking the main elements of our formal framework to the main GRL elements. The part below the dashed horizontal line depicts GRL elements. A GRL Diagram (bottom) contains zero or more IntentionalElements, either a Goal, a Softgoal, a Task, or a Belief. An ElementLink is either a Contribution, a Decomposition, or a Dependency and contains an IntentionalElements as source and as target.

The part above the horizontal dashed line depicts the concepts we introduced in the previous section. The top-left element Argument (see Section III-B) can attack other arguments (Section III-C) and generalizes both a Formula and an Inference. That is, both these elements can be arguments. A formula has an AcceptStatus (Section III-D). An Inference is from a set of Arguments as premise and a Formula as conclusion. The InferenceType is either strict or defeasible (Section III-A). A Formula is either a Modality (modal formula), a Proposition, a BinaryOperation, or a Negation (negated proposition). The Modality can be either B (belief), G (goal), E (evidence), or A (action). A BinaryOperation is either a Disjunction, a Conjunction, or a *contributes_to* Connective (III-A).

Finally, the red arrows depict how the two metamodels are integrated. The left arrow connects a GRL IntentionalElement with an argumentation Modality, where the mapping is denoted with red text. Thus, an intention element traces to an argument, which is always a modal formula. The right arrow connects the GRL ElementLink with the argumentation Connective (i.e. a *contributes_to* connective). Thus, an arrow between elements of a GRL model corresponds to *contributes_to* connectives in the argumentation framework.

[2]Note that in A_2 we use an unspecified defeasible inference rule saying roughly that 'if something costs money and we want to increase our profits, we should not do it'.

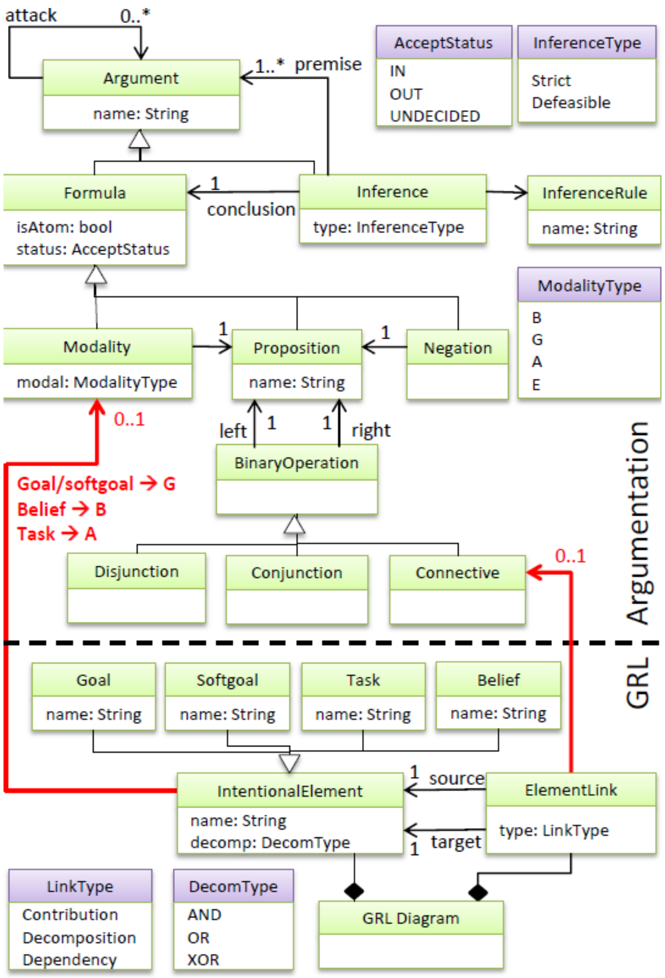

Fig. 6: The Metamodel

V. RESEARCH AGENDA

A. Extension, Implementation and Evaluation

The current paper briefly describes how formal argumentation could be used to capture discussions about goals and evidence. The first objective of our future work is to expand on this basis by, for example, capturing specific argumentation patterns and critical questions for evidence-based requirements engineering in our formal framework (cf. [3], [5]). URN has been further extended with, for example, LEGAL-URN [9], which deals with legal compliance. Through the integration with the argumentation framework, organizations can then capture the evidence for selecting different alternatives interpretation of legal text [10], and this evidence can in turn be presented to the auditor as a proof of compliance.

GRL has an open source Eclipse-based tool support called jUCMNav [2]. Based on the metamodel of the previous section, import-export functions can be defined in jUCMNav, which, for example, allow us to convert practical reasoning arguments from the design discussions to a GRL model. These arguments can be built using the Argument Web tools [4]. For example, OVA[3] can be used to quickly build arguments such as the ones in Figures 2 and 3, and TOAST[4] can be used to evaluate the acceptability of these arguments. The Argument Web is based on a standard and well-defined argument ontology and include web services for importing and extracting argument data in a JSON linked data format.

In order to demonstrate the applicability of our approach and to evaluate our method, we aim to complete our implementation and test factors such as scalability and usability. This will also allow us make the connection to the practical aspects of the requirements engineering. One of the core questions here is what the benefits of using argumentation are compared to not having a formal framework and, for example, keeping information about rationalizations and evidence in unstructured comments.

B. Rationalization of Use Case Maps

Recall from the introduction that GRL is part of URN, an ITU-T standard which combines goals (modeled by GRL) and scenarios (modeled by Use Case Maps (UCM)) in one single notation. Although we have primarily focused on GRL in this paper, we aim in future work to extend our analysis to UCM as well. UCM is used to model scenarios, business processes and functional requirements, and provides means for reasoning about scenarios by establishing *URN links* between intentional elements (such as softgoals and goals) in GRL and non-intentional elements in UCM. Thus, with capturing evidences and rationale behind alternatives, it is possible to reason about the different business processes and scenarios in the organization.

We believe that this a good fit with the work on stories and argumentation. In [6], a hybrid formal theory is introduced that allows one to reason about different stories using arguments. The stories are used to causally explain the most important facts of a case and arguments based on evidence are used to support and attack these stories and each other. Since these stories are very similar to use case maps, we can apply the same techniques to reason about the plausibility of different use case maps, based on underlying goal models, which are in turn rationalized by arguments and evidence.

VI. RELATED WORK

Several evaluation algorithms have been developed to compute the satisfaction level of high-level fuzzy goals based on the selection of lower level goals and tasks [2]. Gross and Yu [11] explore the use of goal-oriented approaches to provide links between business goals, architectural design decisions and structure in a systematic way. They also provide a qualitative evaluation algorithm for goals. However, as mentioned before, we assess not the *satisfiability* of goals given certain tasks that are performed, but rather the *acceptability* of the goal model itself given the discussion between the stakeholders.

Argumentation has been applied to requirements engineering in other work. For example, both Haley *et al.* [12] and

[3]http://ova.arg-tech.org/
[4]http://toast.arg-tech.org/

Franqueira *et al.* [8] use structured arguments to capture design rationales and to show that a system can satisfy its security requirements, respectively. The argumentative part of their work, however, does not include formal semantics for determining the acceptability of arguments. Furthermore, while arguments are included in the design process, there is no explicit trace from arguments to goals tasks. Jureta *et al.* [14] propose a method to guide the justification of goal modeling choices, and include a detailed formal argumentation model that links to goal models. Although their formalization is comparable to ours, we believe that our work paves the way for further enrichment, standardization and implementation of argumentation in goal models. Our argumentation method will draw from both [3] and [5] in order to provide detailed techniques for reasoning with evidence about goals and actions, such as critical questions designed to probe and assess goal-based and evidential reasoning. GRL is part of an accepted standard [13], and $ASPIC^+$ has become a mainstay in AI research on computational argumentation, fueled also by its well-defined connections to Dung's semantics [7] and the Argument Web [4]. Finally, both GRL and $ASPIC^+$ have existing implementations available, which facilitates further tool development. We have presented initial results of this paper in [17].

VII. Conclusion

We propose a framework for traceability of GRL elements to the arguments and evidence of the stakeholders. First, we extend the $ASPIC^+$ framework for formal argumentation [15] with rules for practical and evidential reasoning. This allows us to formally capture evidence-based discussions between stakeholders and subsequently determine the acceptability status of arguments' conclusions (e.g. goals, beliefs) using argumentation semantics [7]. Next, we integrate the formal argumentation model with GRL elements using a UML metamodel. Arguments in our framework are built from propositions, which are either goals, actions, connectives, beliefs, or evidence. These elements correspond respectively to softgoals and goals, tasks, links between intentional elements, and beliefs. Thus, we can compute the acceptability of a GRL model given the evidence and design arguments given by the stakeholders. Finally, we set out a research agenda describing future work consisting of an extension to argumentation patterns and critical questions, an extension to LEGAL-URN, an implementation to evaluate our framework, and the rationalization of Use Case Maps.

Acknowledgements

This work has been partially funded by AFR - PDR grant #5810263 and the National Research Fund (FNR), Luxembourg, RationalArchitecture project.

References

[1] D. Amyot. Introduction to the user requirements notation: Learning by example. *Computer Networks*, 42(3):285–301, June 2003.

[2] D. Amyot, S. Ghanavati, J. Horkoff, G. Mussbacher, L. Peyton, and E. S. K. Yu. Evaluating goal models within the goal-oriented requirement language. *International Journal of Intelligent Systems*, 25:841–877, August 2010.

[3] K. Atkinson and T. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171(10):855–874, 2007.

[4] F. Bex, J. Lawrence, M. Snaith, and C. Reed. Implementing the argument web. *Communications of the ACM*, 56(10):66–73, 2013.

[5] F. Bex, H. Prakken, C. Reed, and D. Walton. Towards a Formal Account of Reasoning about Evidence: Argumentation Schemes and Generalisations. *Artificial Intelligence and Law*, 11(2/3):125–165, 2003.

[6] F. Bex, P. van Koppen, H. Prakken, and B. Verheij. A hybrid formal theory of arguments, stories and criminal evidence. *Artificial Intelligence and Law*, 2:123–152, 2010.

[7] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[8] V. N. Franqueira, T. T. Tun, Y. Yu, R. Wieringa, and B. Nuseibeh. Risk and argument: A risk-based argumentation method for practical security. In *19th IEEE International Requirements Engineering Conference*, USA, June 2011. IEEE.

[9] S. Ghanavati. *Legal-urn framework for legal compliance of business processes*. PhD thesis, University of Ottawa, 2013.

[10] S. Ghanavati and H. Joris. Impact of legal interpretation in business process compliance. In *TELERISE @ICSE*, 2015.

[11] D. Gross and E. Yu. Evolving system architecture to meet changing business goals: an agent and goal-oriented approach. In *Proc. 5th IEEE Int. Symposium on RE*, pages 316–317, 2001.

[12] C. B. Haley, R. Laney, J. D. Moffett, and B. Nuseibeh. Security requirements engineering: A framework for representation and analysis. *Software Engineering, IEEE Transactions on*, 34(1):133–153, 2008.

[13] ITU-T. Recommendation Z.151 (11/08): User Requirements Notation (URN) – Language Definition. http://www.itu.int/rec/T-REC-Z.151/en, 2008.

[14] I. J. Jureta, S. Faulkner, and P.-Y. Schobbens. Clear justification of modeling decisions for goal-oriented requirements engineering. *Requir. Eng.*, 13(2):87–115, May 2008.

[15] S. Modgil and H. Prakken. A general account of argumentation with preferences. *Artificial Intelligence*, 195:361–397, 2013.

[16] A. Van Lamsweerde. Goal-oriented requirements engineering: A guided tour. In *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*, pages 249–262. IEEE, 2001.

[17] M. van Zee and S. Ghanavati. Capturing evidence and rationales with requirements engineering and argumentation-based techniques. In *Proceedings of the 26th Benelux Conference on Artificial Intelligence (BNAIC2014)*, November 2014.